10. ročník
Žilina 2. – 4. 7. 2018
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
ossconf.soit.sk
OSS Conf

OSSConf 2025: 1–8

První kontakt s programem Manim

PAVEL STŘÍŽ (CZ)

Abstrakt. Tento článek tvoří úvod do práce s programem Manim (matematické animace) založeném na Pythonu s podporou TEXu. Jedná se především o rešerši zdrojů, tipy, triky a řešení některých problematických partií. Program vytváří a spravuje Grant Sanderson alias 3blue1brown a Ben Eater. Původně to byl soukromý projekt pomáhající jim programově vytvořit náročnější videa, nyní se jedná o otevřený software.

Klíčová slova. Animace, Manim, Python, TEX.

THE FIRST ENCOUNTER WITH Manim

Abstract. This article is an introduction to work with Manim software (Mathematical Animation Engine), which is a Python-based program with support of TEX. The paper consists mainly of research of sources, tips, tricks and solution to some problematic parts. Software is being developed and maintained by Grant Sanderson alias 3blue1brown and Ben Eater. It was initially a private project supporting them programatically create complex videos. Now, it's an open-source software.

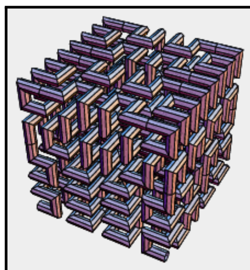
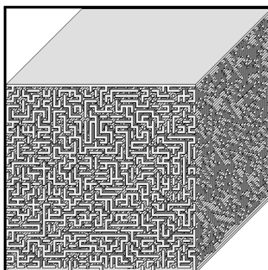
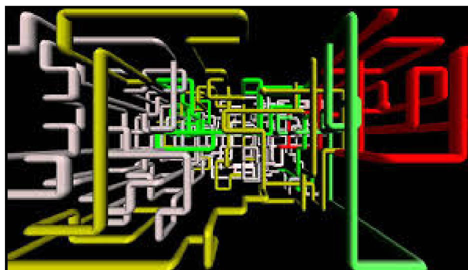
Keywords. Animation, Manim, Python, TEX.

Motto: Python se má rád, je to tam jedno `selfí` za druhým!

1. Mohou za to kvaterniony

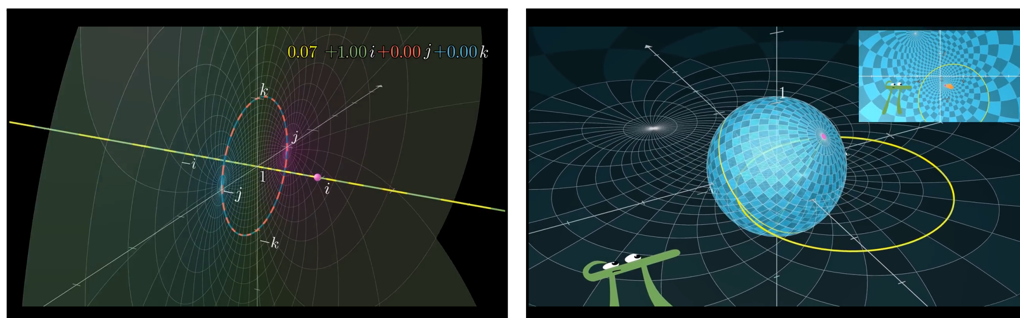
Někdy v roce 2018 jsem otevřel jeden ze starých problémů, a to jak vykreslit nekonečnou trubku, ale tak, aby se vzájemně nekřížila. Je to trochu obdoba Hilbertovy křivky (vlevo) či tvorba bludišť, např. v programu Daedalus (uprostřed). Zájemce odkazuji na demo (Xubuntu 20.04, obrázek vpravo):

```
$ sudo apt install xscreensaver xscreensaver-gl
$ xscreensaver &
$ sleep 8; xscreensaver-command -demo 39
```

Podařilo se mi to vyřešit v Blenderu, ale tehdy dokumentace ke kvaternionům (rozšíření komplexních čísel pro 3D) byla strohá, hledal jsem doplňující materiály. Zaujalo mě video `youtube.com/watch?v=d4EgbgTm0Bg` s detaily na `eater.net/quaternions`, ale co víc, na fóru byla zmínka, že video bylo vytvořené v programu Manim. Tak jsem se do toho víc ponořil, neb mi název programu nic neříkal.

Zde je pár ukázek od tvůrců ze zmíněného videa:

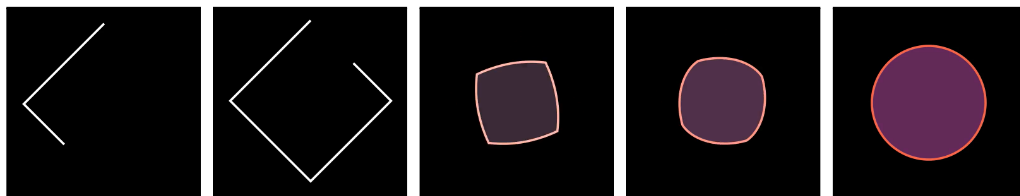


2. Hello World! aneb Manim animuje čtverec na kolečko

Jedná se o vyvíjený program na matematické animace a je docela programátorský oříšek dát vše do latě (instalaci, rozběhnutí ukázek, vlastní tvorba). Bral jsem to však za součást učení se.

Program lze nainstalovat přes `pip3` (`manimlib`), `virtualenv`, `docker`, `anaconda`, ale hlavně přímo. Detaily jsou na `github.com/3b1b/manim`.

```
$ git clone https://github.com/3b1b/manim.git
$ cd manim
$ sudo -H pip3 install -r requirements.txt
$ python3 manim.py example_scenes.py SquareToCircle -pl
```



Pokud vše proběhne v pořádku spustí se vám video, kdy se bílý čtverec změní na barevné kolečko a to pak zmizí.

Problém jsem měl s instalací knihovny `pycairo`, to jsem vůči manuálu vyřešil přes:

```
$ sudo apt install python3-cairo
$ sudo -H pip3 install manimlib --ignore-installed pycairo
```

Druhý řádek mi v roce 2020 nejel, stále se mi snažil vnutit pycairo, a to při instalaci spadne, tak jsem užil 1. řádek z předchozího skriptu, v requirements.txt jsem zakomentoval řádek s pycairo a doinstaloval jen další závislosti:

```
$ sudo -H pip3 install -r requirements.txt
```

Touto cestou užívám starší pycairo verze 1.16.2 a Manim z pracovního adresáře, vše dále představené běží.

3. Hlouběji u spuštění z příkazového řádku

Nápovědu lze získat přes python3 -m manim --help. Ukázky jsou realizovány přes třídy, jejich seznam jsem si nechal vypsat přes cat <soubor.py> | grep <třída či class>. Obvykle vidíme Scene, GraphScene, ThreeDScene a SVGMobject.

Vytvoření videa se realizuje přes:

```
python3 -m manim <soubor.py> [třídy] [parametry]
```

Třídy oddělujeme mezerami. Bez zadání třídy vyběhne nabídka a zadáváme čísla oddělená čárkou. Ne všechny třídy jsou takto spustitelné.

Nejběžnější volitelné parametry jsou následující:

- -p = preview, otevření souboru po dokončení,
- -l | -m | bez parametru = velikost videa, low | medium | high,
- -t = transparent, pozadí bude průhledné,
- -c = color, pozadí bude mít specifickou barvu,
- -r = resolution, rozlišení v pixelech, výška čárka délka,
- -s = save the last frame, uloží z videa jen poslední snímek,
- --livestream | --to-twitch = živé vysílání,
- -h | --help = nápověda.

Výsledky se ukládají do adresáře media/video, pak do složky dle názvu skriptu, následuje složka výška p rychlost, např. 480p15 znamená 480 pixelů je výška obrazu v rychlosti 15 snímků/vteřinu. Vzniká série malých videí, které se v závěru spojí do velkého souboru mp4.

Generované TEXové soubory, xdv a svg lze nalézt v adresáři media/TeX.

4. Inspirativní zdroje

Mezi tutoriály v angličtině řadíme talkingphysics.wordpress.com/2019/01/08/getting-started-animating-with-manim-and-python-3-7 s podpůrnými soubory na github.com/zimmermant/manim_tutorial. Inspirací jsou i zdrojové kódy na github.com/Solara570/demo-solara.

Další tutoriál lze nalézt na github.com/malhotra5/Manim-Tutorial, obsáhlý v čínštině pak na github.com/cai-hust/manim-tutorial-CN.

Kdo dává přednost videotutoriálům, nechť nahlédne např. na `https://www.youtube.com/channel/UCxiWCEdx7aY88bSEUgLOC6A` s podpůrnými soubory na `github.com/Elteoremadebeethoven/AnimationsWithManim`.

Za poklad ke zkoumání však lze považovat přímo ve složce `manim` adresář `from_3b1b`, speciálně adresář `old`. Řada kódů však nejede na první dobrou, neb autoři Manim přepracovávají a upravují. Nechávám otevřené pro badatele, je to spleť překlepů, programátorských háků a háčků v Pythonu.

V češtině v akademickém roce 2017/2018 proběhl seminář od Mirka Olšáka, `http://www.olsak.net/mirek/manim/`, od 3/2020, možná v souvislosti s koronavirem, vznikají překlady anglických videí, `https://www.youtube.com/channel/UCIhWS2rX78OXidZ87QLkkxA`.

5. Otázky a odpovědi

Představím některé běžné situace a jejich řešení.

5.1. Jak řešit diakritické znaky?

Jinými slovy, lze zasáhnout do preambule TEXové šablony? Šablona je uložena v souboru `manimlib/tex_template.tex`, po záloze souboru jsem jej upravil do této podoby:

```
\documentclass[preview]{standalone}
\usepackage[czech]{babel}
\usepackage[utf8]{luainputenc}
\usepackage[IL2]{fontenc}
\usepackage{amsmath,amssymb,halloweenmath,tikz,fdsymbol}
\renewcommand\rmdefault{cmr} \renewcommand\sfdefault{cmss}
\renewcommand\ttdefault{cmtt}
\begin{document}
YourTextHere
\end{document}
```

V pozadí se užívá `latex` s převodem do svg. Přepnul jsem si na `lualatex` s jedním parametrem navíc v `manimlib/utils/tex_file_writing.py`:

```
        commands = [
            "lualatex",           # byl "latex",
            "-output-format=dvi", # přidaný řádek
```

Náš kód v souboru `ukazky/ahoj-svete.py` by mohl vypadat takto:

```
from manimlib.imports import *
class AhojSvete(Scene):
    def construct(self):
        svetu = TextMobject("Ahoj, světe!")
        self.play(Write(svetu))
        self.wait()
```

Spouštíme: `python3 -m manim ukazky/ahoj-svete.py -p`



5.2. Lze zařadit TikZ?

Lze, ale síla linek se v svg nebere v potaz. Chce to experimentovat u konkrétního kódu. Zde je ukázka animace pro divadelní hru v souboru `ukazky/moda-live.py`, která nebyla použita, sloužila k testovacím účelům obrazovek a monitorů. Je vidět složitější animace, vsunutí textu do TikZu, u proměnných užívám `r""`, abych nemusel zdvojovat zpětná lomítka.

```
from manimlib.imports import *
class ModaLive(Scene):
  def construct(self):
    text=r"\begin{tikzpicture}\node[draw=none]{Móda live!};\end{tikzpicture}";
     modalive=TextMobject(text); modalive.scale(6); obdelnik=[""]*5
    obdelnik[0]=TextMobject(text); obdelnik[0].scale(2); obdelnik[0].to_edge(DOWN);
     obdelnik[0].set_color(YELLOW); obdelnik[0].rotate(PI)
    obdelnik[1]=TextMobject(text); obdelnik[1].scale(2); obdelnik[1].to_edge(UP);
     obdelnik[1].set_color(YELLOW)
    obdelnik[2]=TextMobject(text); obdelnik[2].scale(2); obdelnik[2].to_edge(LEFT);
     obdelnik[2].set_color(YELLOW); obdelnik[2].rotate(PI/2)
    obdelnik[3]=TextMobject(text); obdelnik[3].scale(2); obdelnik[3].to_edge(RIGHT);
     obdelnik[3].set_color(YELLOW); obdelnik[3].rotate(-PI/2)
    self.wait(1); self.play(Write(modalive),run_time=20); self.wait(1);
     self.play(FadeOut(modalive),run_time=5); self.wait(1)
    for x in [1,3,0,2]: self.play(Write(obdelnik[x]), run_time=2)
    modalive.set_color_by_gradient(RED,BLUE); self.wait(1);
     self.play(GrowFromCenter(modalive), run_time=5); self.wait(1);
     self.play(Transform(obdelnik[0], modalive),Transform(obdelnik[1], modalive),
     run_time=3); self.wait(1); self.play(Transform(obdelnik[2], modalive),
     Transform(obdelnik[3], modalive), run_time=3)
    for x in range(4): self.remove(obdelnik[x])
    self.play(FadeOut(modalive), run_time=5); self.wait(1);
     srdce=TextMobject(r"$\varheartsuit$"); srdce.scale(20); srdce.set_color(RED);
     srdce.rotate(-45); self.play(GrowFromCenter(srdce), run_time=5);
     self.play(FadeOut(srdce), run_time=2); self.wait(1);
     self.play(ShowCreation(modalive), run_time=40); self.wait(1);
     self.play(FadeOut(modalive), run_time=5); self.wait(1)
```

Animace: `python3 -m manim ukazky/moda-live.py -p`



5.3. Lze animovat matematiku?

Pozornější TEXisté si všimli, že v preambuli šablony je balíček halloweenmath, nyní jej použijeme v souboru `ukazky/rovnice.py`.

```
from manimlib.imports import *
class Rovnice(Scene):
  def construct(self):
    sum1=TextMobject(r"$$\sum_{i=0}^{n} {n\choose i} a^{n-i} b^i = (a+b)^n$$")
    sum1.scale(2); sum1.to_edge(UP); sum1.set_color(WHITE); self.play(Write(sum1),
     run_time=3); self.wait(1)
    sum2=TextMobject(r"$$\mathwitch*_{i=0}^{n} {n\choose i} \mathrightghost^{n-i}
     \mathleftghost^i \xrightwitchonbroom*{} (\mathrightghost+\mathleftghost)^n$$")
    sum2.scale(1.75); sum2.to_edge(DOWN); sum2.set_color(WHITE)
    self.play(ReplacementTransform(sum1.copy(), sum2), run_time=6); self.wait(4)
```

Animace: `python3 -m manim ukazky/rovnice.py -p`



5.4. Lze animovat kandži?

Asi nejrychlejší způsob je si v `manimlib/constants.py` zapnout ChineseTEX:

```
TEX_USE_CTEX = True # False
```

Tím si zajistíme, že budeme užívat `xelatex` a TEXovou šablonu v souboru `manimlib/ctex_template.tex`. Tu jsem si po odzálohování upravil do této podoby:

```
\documentclass[preview]{standalone}
\usepackage{amsmath,amssymb}
\usepackage[UTF8]{ctex}
\begin{document}
YourTextHere
\end{document}
```

Jednoduchá ukázka by mohla vypadat takto:

```
from manimlib.imports import *
class Japonstina(Scene):
  def construct(self):
    sum1=TextMobject(r"今日は何曜日ですか。")
    sum1.scale(2.5); sum1.to_edge(UP); sum1.set_color(RED); self.play(Write(sum1),
     run_time=3); self.wait(1)
    sum2=TextMobject(r"３日です。")
    sum2.scale(6); sum2.to_edge(DOWN); sum2.set_color(RED)
    self.play(ReplacementTransform(sum1.copy(),sum2),run_time=5); self.wait(2)
```

Animaci získáme přes: `python3 -m manim ukazky/japonstina.py -p`



5.5. Lze animovat výstupy z teorie grafů?

Na pomoc jsem si vzal knihovnu `manimnx` a vypnul jsem si užití CTeXu:

```
TEX_USE_CTEX = False
```

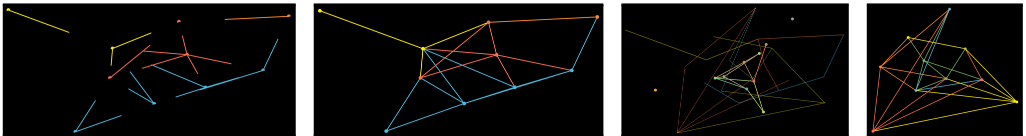
Doinstaloval jsem si potřebné:

```
$ git clone https://github.com/rajatvd/manimnx
$ sudo -H pip3 install networkx==2.3
```

V `manimnx/example.py` jsem zasáhl do jednoho řádku tímto způsobem, protože `manim.py` načítám z pracovního adresáře:

```
import manimnx.manimnx.manimnx as mnx
```

Ukázka: `python3 -m manim manimnx/example.py RandomGraphs -p`



5.6. Lze animovat diagramy?

Jako poslední ukázku jsem vybral extrémní případ knihovny `Danim`.

```
$ git clone https://github.com/graviton1221/Danim
$ sudo -H pip3 install pandas
```

V `Danim/BubbleChart/bubblechart_constant.py` jsem \\ upravil na /, jak pracuji pod Linuxem. A ještě jednou jsem si zapnul CTeX.

Každý popisek se generuje do zvláštního TeXového souboru, vznik animace tedy trvá: `python3 -m manim Danim/BubbleChart/BubbleChartAnimation.py BubbleChartAnimation -p`

Zde je ukázka z 19. vteřiny videa.

6. Co s programem Manim neumím?

Co jsem nepotřeboval a nezkoušel jsem do hloubky je živý přenos přes protokol tcp (parametr `--livestream`) s podporou pro Twitch přes parametry `--to-twitch` a `--with-key`. Hashovací klíč jsem našel na `twitch.tv` pod uživatelem; Settings; Channel and Videos a pak Primary Stream key (Copy nebo Show).

Práce se soubory svg je nativní, zde jsou oblíbené figurky přednášejících (PíCreature, Stickman a Linus). Lze si vytvořit vlastního průvodce. Nezkoušel jsem, byť návod existuje, `https://talkingphysics.wordpress.com/2018/08/14/working-with-svg-files-manim-series-part-12/`.



Vzniká podpora pro JavaScript, viz `github.com/JazonJiao/Manim.js`. Přes JavaScript bych na animace volil spíš `d3js.org`.

Vzniká též webové rozhraní, viz `github.com/eulertour/eulerv2`. To mi přišlo ve webovém prohlížeči extrémně pomalé.

Pokud chcete nahlédnout na animace zmíněné v článku, a programátorské a TEXové věci si nezkoušet, navštivte můj skromný kanál na YouTube:

`youtube.com/playlist?list=PLnD-4Ssyh8yOQi08n9L8WJ3wV3u2p9VO9`

Co zmínit závěrem? Je to jiný svět, snad jsem vám hlavu nezamotal víc než je zdrávo, mám totiž před sebou víc otázek než odpovědí, i tak však přeji hezké bádání s nástrojem Manim!

- Můžeme užít LuaLATEX a libovolné písmo přes balíček fontspec?
- Lze získat animaci jako sérii souborů svg?
- Může se objekt pohybovat po libovolné křivce, např. po spirále?
- Je možné (de)aktivovat vyhlazování písem?
- Jak lze ideálně zařadit grafiku z programů METAPOST a PSTricks?
- Jak by to bylo se zařazením 3D objektů (Asymptote, Blender)?
- Lze vložit grafické výstupy z programů typu Matplotlib a R?
- Lze vložit do vznikající animace video? Například ve videu o kvaternionech je pohyb ruky, tedy to lze, ale jak …

Je vidět, že 3blue1brown je učitel tělem i duší a nutí nás přemýšlet a programátorsky hledat a experimentovat. Ne nadarmo byl 13. 3. 2020 Grant Sanderson pozván jako řečník do TEDxBerkeley (`https://youtube.com/watch?v=s_L-fp8gDzY`) s tématem *What Makes People Engage With Math*, minimálně jako duševní kantorská podpora v době koronaviru.

… if you have a soul, you have to know why, right?
— Grant Sanderson @ 00:13:04,159265

Kontaktní adresa

Ing. Pavel Stříž, Ph.D., U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: `pavel@striz.cz`